Year: 2012

Modelling and optimizing the process of learning mathematics

Käser, Tanja ; Busetto, Alberto Giovanni ; Baschera, Gian-Marco ; Kohn, Juliane ; Kucian, Karin ; Aster, Michael ; Gross, Markus

DOI: https://doi.org/10.1007/978-3-642-30950-2_50

Originally published at:
Käser, Tanja; Busetto, Alberto Giovanni; Baschera, Gian-Marco; Kohn, Juliane; Kucian, Karin; Aster, Michael; Gross, Markus (2012). Modelling and optimizing the process of learning mathematics. In: ITS'12 Proceedings of the 11th international conference on Intelligent Tutoring Systems, Chania, Crete, Greece, 14 June 2012 - 18 June 2012, 389-398.
DOI: https://doi.org/10.1007/978-3-642-30950-2_50

Modelling and Optimizing the Process of Learning Mathematics

Tanja Käser[1], Alberto Giovanni Busetto[1,2], Gian-Marco Baschera[1], Juliane Kohn[4], Karin Kucian[3], Michael von Aster[3,4,5], and Markus Gross[1]

[1] Department of Computer Science, ETH Zurich, Zurich, Switzerland
[2] Competence Center for Systems Physiology and Metabolic Diseases, Zurich, Switzerland
[3] MR-Center, University Children's Hospital, Zurich, Switzerland
[4] Department of Psychology, University of Potsdam, Potsdam, Germany
[5] Department of Child and Adolescent Psychiatry, German Red Cross Hospitals Westend, Berlin, Germany

Abstract. This paper introduces a computer-based training program for enhancing numerical cognition aimed at children with developmental dyscalculia. Through modelling cognitive processes and controlling the level of their stimulation, the system optimizes the learning process. Domain knowledge is represented with a dynamic Bayesian network on which the mechanism of automatic control operates. Accumulated knowledge is estimated to select informative tasks and to evaluate student actions. This adaptive training environment equally improves success and motivation. Large-scale experimental data quantifies substantial improvement and validates the advantages of the optimized training.

Keywords: learning, control theory, optimization, dynamic Bayesian network, dyscalculia.

1 Introduction

Computer-assisted learning is gaining importance in children's education. Intelligent tutoring systems are successfully employed in different fields of education, particularly to overcome learning disabilities [1]. The application of computers extends conventional learning therapy. This study presents a computer-based training program for enhancing numerical cognition, aimed at children with developmental dyscalculia (DD) or difficulties in learning mathematics. It entertains the idea that the learning process can be optimized through modelling cognitive development and control.

Motivation. DD is a specific learning disability affecting the acquisition of arithmetic skills. Genetic, neurobiological, and epidemiological evidence indicates that DD is a brain-based disorder with a prevalence of 3-6% [2]. Challenges are subject-dependent and hence individualization is needed to achieve substantial improvements. Computer-based approaches enable the design of adaptable training, by estimating abilities and by providing intensive training in a stimulating environment. The learner gains self efficacy and success, in turn leading to increased motivation.

Related work. Previous studies evaluated computer-based trainings for number processing and calculation, documenting promising results [3–5]. Available trainings are designed specifically for children with DD, yet provide limited user adaptation. In the domain of mathematics, intelligent tutoring systems focus on specific aspects of the domain [6–8]. A plethora of advanced control approaches aimed at optimization of complex mechanisms exists in the literature [9]. As in this study, controllers can be based upon explicit models obtained through intervention-driven identification [10]. Related predictive models aimed at treating learning disabilities have been introduced for spelling learning [1, 11].

Contribution. We model the cognitive processes of mathematical development. Recent neuropsychological findings are incorporated into a predictive dynamic Bayesian network. We introduce automatic control aimed at optimizing learning. This model predictive control enables a significant level of cognitive stimulation which is user- and context-adaptive. Results from two large user-studies quantify and validate the improvements induced by training.

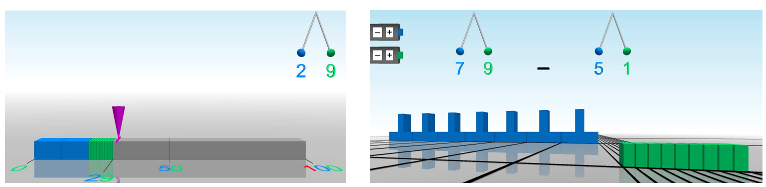
2 Training environment

Current neuropsychological models postulate the existence of task-specific representational modules located in different areas of the brain. The functions of these modules are relevant to both adult cognitive number processing and calculation [12]. Dehaene's triple-code model [13] presumes three representational modules (verbal, symbolic, and analogue magnitude) related to number processing. These modules develop hierarchically over time [14] and the overlap of the number representations increases with growing mathematical understanding [17]. The development of numerical abilities follows a subject-dependent speed which is influenced by the development of other cognitive as well as domain general abilities and biographical aspects [14]. Hence, when teaching mathematics, a substantial degree of individualization may not only be beneficial, but even necessary. The introduced computer-based training addresses these challenges by

1. structuring the curriculum on the basis of natural development of mathematical understanding (hierarchical development of number processing).
2. introducing a highly specific design for numerical stimuli enhancing the different representations and facilitating understanding. The different number representations and their interrelationsships form the basis of number understanding and are often perturbed in dyscalculic children [14].
3. training operations and procedures with numbers. Dyscalculic children tend to have difficulties in acquiring simple arithmetic procedures and show a deficit in fact retrieval [15, 16].

4. providing a fully adaptive learning environment. Student model and controlling algorithm optimize the learning process by providing an ideal level of cognitive stimulation.

Structure of the training program. The training is composed of multiple games in a hierarchical structure. Games are structured according to number ranges and further grouped into two areas. The first area focuses on "number representations and understanding". It trains the transcoding between alternative representations and introduces the three principles of number understanding: cardinality, ordinality, and relativity. Games in this area are structured according to current neuropsychological models [13, 14]. The first area is exemplified by the LANDING game (Fig. 1(a)). The second area is that of "cognitive operations and procedures with numbers", which aims at training concepts and automation of arithmetical operations. This is illustrated by the PLUS-MINUS game (Fig. 1(b)). Games are divided into main games requiring different abilities and support games training specific ones, serving as basic prerequisites. Difficulty estimation and hierarchy result from the development of mathematical abilities.



(a) LANDING game. (b) PLUS-MINUS game.

Fig. 1. In the LANDING game, the position of the displayed number (29) needs to be indicated on the number line. In the PLUS-MINUS game, the task displayed needs to be modeled with the blocks of tens and ones.

Design of the numerical stimuli. Properties of numbers are encoded with auditory and visual cues such as color, form, and topology. The digits of a number are attached to the branches of a graph and represented with different colors according to the positions in the place-value system (see left of Fig. 2). Numbers are illustrated as a composition of blocks with different colors, i.e., as an assembly of one, ten and hundred blocks. Blocks are linearly arranged from left to right or directly integrated in the number line (Fig. 2 right). Showing all stimuli simultaneously in each game of the training program reinforces links between different number representations and improves number understanding.

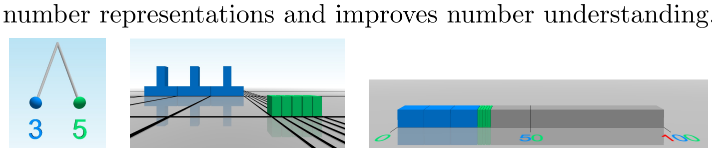


Fig. 2. Design of numerical stimuli for the number 35.

3 Selection of actions

A fundamental component is the pedagogical module: the subsystem making the teaching decisions. It selects the skills for training and determines the actions. The mechanisms adaptively assess user inputs and dynamically optimize decisions [9]. The learner state is estimated and internally represented by the student model. An attached bug library enables recognition of error patterns.

3.1 Student model

The mathematical knowledge of the learner is modelled using a dynamic Bayesian network [18]. The network consists of a directed acyclic graphical model representing different mathematical skills and their dependencies. This representation is ideal for modelling mathematical knowledge as the learning domain exhibits a distinctively hierarchical structure. The resulting student model contains 100 different skills (Fig. 3).The structure of the net was designed using experts' advice and incorporates domain knowledge [13–16]. Two skills s_A and s_B have a (directed) connection, if mastering skill s_A is a prerequisite for skill s_B. The belief of a skill s_{Ai} (probability that skill is in the learnt state) is conditioned over its parents π_i:

$$p(s_{A1}, ..., s_{An}) = \prod_i p_{s_{Ai}} \quad \text{where} \quad p_{s_{Ai}} := p(s_{Ai}|\pi_i) \tag{1}$$

As the skills cannot be directly observed, the system infers them by posing tasks and evaluating user actions. Such observations (E) indicate the presence of a skill probabilistically. The posteriors $p_{s_{Ai}|E_k}$ of the net are updated after each solved task k using the sum-product algorithm (libDAI [19]). Initially, the probabilities are initialized to 0.5 (principle of indifference). The dynamic Bayesian net has a memory of 5, i.e. posteriors are calculated over the last five time steps.

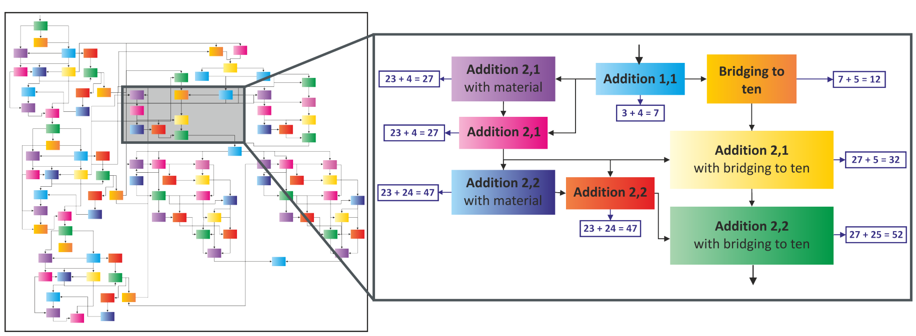


Fig. 3. Skill net containing 100 skills (left), zoom of addition skills from 0-100 (right).

3.2 Controller

The selection of actions is rule-based and non-linear. Rather than following a specified sequence to the goal, learning paths are adapted individually. This increases the set of possible actions (due to multiple precursors and successors). The controller selects one of the following options based on the current state:

1. **Stay**: Continue the training of the current skill;
2. **Go back**: Train a precursor skill;
3. **Go forward**: Train a successor skill;

The decision is based on the posterior probabilities delivered by the student model. After each solved task, the controller fetches the posterior probability $p_{s|E}(t)$ of the skill s being trained at time t. Then, $p_{s|E}(t)$ is compared against a lower and an upper threshold, denoted by $p^l_s(t)$ and $p^u_s(t)$. The resulting interval defines the optimal training level: if the probability lies between the thresholds, 'Stay' is selected. In contrast, 'Go Back' and 'Go forward' are selected when $p_{s|E}(t) < p^l_s(t)$ and when $p_{s|E}(t) > p^u_s(t)$. Thresholds are not fixed: they converge with more played samples (n_c):

$$p^l_s(t) = p^{l0}_s(t) \cdot {l_c}^{n_c} \quad \text{and} \quad p^u_s(t) = p^{u0}_s(t) \cdot {u_c}^{n_c} \tag{2}$$

Initial values of the upper ($p^{l0}_s(t)$) and lower ($p^{u0}_s(t)$) thresholds as well as the change rates (l_c, u_c) are heuristically determined. The convergence of the thresholds ensures a sufficiently large number of solved tasks per skill and prevents training the same skill for too long without passing it.
When 'Stay' is selected, a new appropriate task is built. Otherwise, a precursor (or successor) skill is selected by fetching all precursor (successor) skills of the current skill and feeding them into a decision tree. Figure 4 shows the simplified decision trees for 'Go Back' and 'Go Forward'. The nodes of the trees encode selection rules. If errors matching patterns of the bug library are detected, the relevant remediation skill is trained. If a user fails to master skill s_A and goes back to s_B, s_A is set as a recursion skill. After passing s_B, the controller will return to s_A. To consolidate less sophisticated skills and increase variability, selective recalls are used.

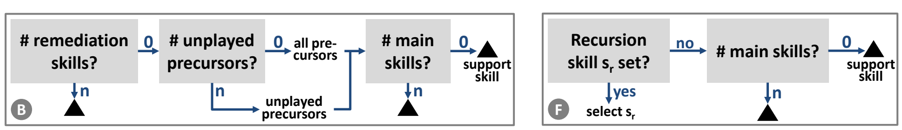


Fig. 4. Decision trees for 'Go Back' (left) and 'Go Forward' (right) options. At the end nodes (triangles), the candidate skill with lowest posterior probability ('Go Back' option)/with posterior probability closest to 0.5 ('Go Forward' option) is selected.

This control design exhibits the following advantages:

1. *Adaptability*: the network path targets the needs of the individual user (Fig. 5).
2. *Memory modelling*: forgetting and knowledge gaps are addressed by going back.
3. *Locality*: the controller acts upon current nodes and neighbours, avoiding unreliable estimates of far nodes.
4. *Generality*: the controller is student model-independent: it can be used on arbitrary discrete structures.

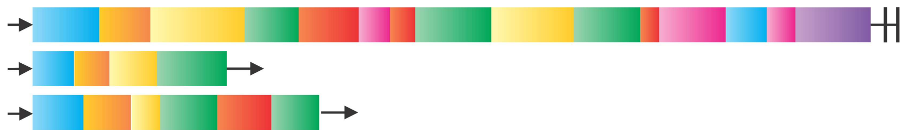


Fig. 5. Skill sequences of three children in addition. Colours are consistent with Fig. 3. User 2 and 3 passed all skills in the range, while user 1 did not pass this range within the training period. The length of the rectangles indicates the number of samples.

4 Methods & Results

Quality of controller and student model have been measured through external effectiveness tests. Experimental data consist of input logs of two on-going large-scale studies (Germany and Switzerland). The studies are conducted using a cross-over design, i.e. participants are divided into a group starting the training immediately and a waiting group. The groups are mapped according to age (2.-5. grade of elementary school), intelligence and gender. All participants visit normal public schools and are German-speaking. They exhibit difficulties in learning mathematics indicated by a below-average performance in arithmetic (addition T-score: 35.4 [SD 7.1], subtraction T-score 35.4 [SD 7.9]) [22]. Participants trained for a period of 6 weeks with a frequency of 5 times per week, during sessions of 20 minutes. Due to technical challenges, a subset of 33 logfiles were completely and correctly recorded. On average, each user completed 29.84 (SD 2.87, min 24, max 36.96) sessions. The total number of solved tasks is 1562 (SD 281.53, min 1011, max 2179), while the number of solved tasks per session corresponds to 52.37 (SD 7.9, min 37.8, max 68.1).

4.1 Logfile analyses

The analyses of the input data show that the participants improved over time. They provide evidence that the introduced control mechanism significantly speeds up the learning process and that it rapidly adapts to the individual user.

Key skills. To facilitate the analysis of the log files, the concept of 'key skills' is introduced. Key skills are defined in terms of subject-dependent difficulty, they are the hardest skills for the user to pass. More formally,

Definition 1. *A skill s_A is a **key skill** for a user U, that is $s_A \in \mathcal{K}_U$, if the user went back to a precursor skill s_B at least once before passing s_A.*

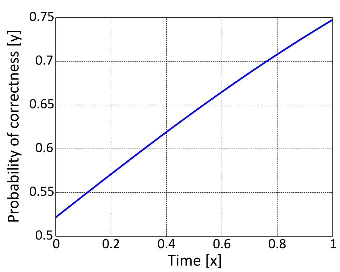
From this follows that the set of key skills $\mathcal{K}_U$ may be different for each user U (and it typically is). In the sequence in Fig. 5, user 2 has no key skills, while user 3 has one key skill (coloured in green) and user 1 has several key skills.

Adaptability of controller. During the study, all participants started the training at the lowest (easiest) skill of the net. The adaptation time $[t_0, t_{\mathcal{K}_U}]$ is defined as the period between the start t_0 of the training and the first time the user hits one of his key skills $t_{\mathcal{K}_U}$. On average, the participants reached their $t_{\mathcal{K}_U}$ after solving 144.3 tasks (SD 113.2, min 10, max 459). The number of complete sessions played up to this point is 1.95 (SD 1.63, min 0.08, max 6.48). These results show, that the model rapidly adjusts to the state of knowledge of the user.

Improvement analysis. To quantify improvement, the learning rate over $\mathcal{K}_U$ is measured from all available samples (both if the participant mastered them during training or not). The improvement over time $I([t_{\mathcal{K}_U}, t_{\text{end}}])$ is computed using a non-linear mixed effect model (NLME) [20] employing one group per user and key skill:

$$y_i \sim \text{Binomial}(1, p_i) \text{ with } p_i = \frac{1}{1 + e^{-(b_0 + b_1 \cdot x_i + u_i)}} \text{ and } u_i \sim \mathcal{N}(0, \sigma^2) \quad (3)$$

where u_i denotes the noise term, x_i the normalized sample indices ($x_i \in [0, 1]$) and y_i the sample correctness. The resulting model (Fig. 6) exhibits an estimated mean improvement of 22.6% (95% confidence interval = [0.21 0.24]).



	$\mathbf{b_0}$	$\mathbf{b_1}$
Estimate(SD)	0.09 (0.06)	1.0 (0.06)
sig.	0.16	<1e-4
95% ci	[-0.073 0.21]	[0.89 1.11]

Fig. 6. The percentage of correctly solved tasks (of key skills) increases over the training period by 22.6% (left side). Exact coefficients of NLME along with standard deviation (in brackets) are plotted by respective significance (sig.) and confidence intervals (ci).

Further analysis demonstrates that the possibility to go back to easier (played or unplayed) skills yields a substantially beneficial effect. The user not only immediately starts reducing the rate of mistakes, but also learns faster. The log files recorded 533 individual cases of going back. All cases in which users play a certain skill (samples x_b), go back to one or several easier skills, and finally pass them to come back to the current skill (samples x_a) are incorporated. Per each case k the correct rate over time $c_{a,k}$ ($c_{b,k}$) is estimated separately for x_a and x_b. Fitting is performed via logistic regression using bootstrap aggregation [21] with resampling ($B = 200$). The direct improvement d_k is the difference between the initial correct rate $c_{a,k}$(at $x_a = 0$) and the achieved correct rate $c_{b,k}$(at $x_b = 1$). The improvement in learning rate r_k is the difference in learning rate over $c_{a,k}$ and $c_{b,k}$. The distributions over $\bar{d}$ (mean over d_k) and $\bar{r}$ (mean over r_k) are well approximated by a normal distribution (Fig. 7). Both measurements are positive on average and a two-sided t-test indicates their statistically significant difference from zero (Tab. 1).

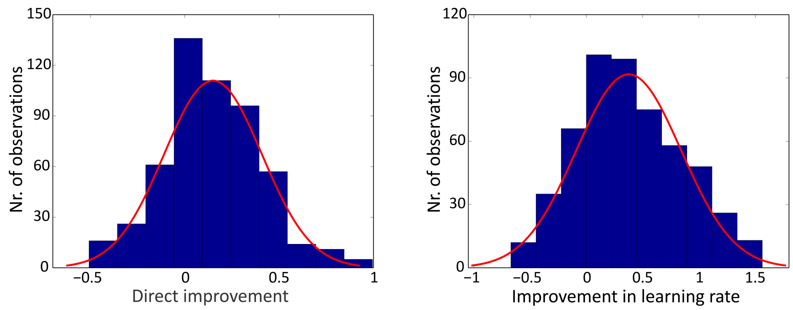


Fig. 7. Distributions over direct improvement $\bar{d}$ and improvement in learning rate $\bar{r}$.

	Mean μ	**sig.**	**99% ci of** μ	**SD** σ	**99% ci of** σ
$\bar{\mathbf{d}}$	0.1494	<1e-6	[0.1204 0.1784]	0.2593	[0.2403 0.2814]
$\bar{\mathbf{r}}$	0.3758	<1e-6	[0.3236 0.4280]	0.4662	[0.4319 0.5059]

Table 1. Statistics for the improvement after going back: Mean improvement μ, significance of mean (sig.), standard deviation (SD), and confidence intervals (ci).

4.2 Training effects

Training effects were measured using external paper-pencil and computer tests. The **HRT** [22] is a paper-pencil test. Children are provided with a list of addition (subtraction) tasks ordered by difficulty. The goal is to solve as many tasks as possible within a time frame of 2 minutes. The **AC** (arithmetic test) exists in a paper-pencil and a computer-based version. Children solve addition (and subtraction) tasks ordered by difficulty. Tasks are presented serially in a time frame of 10 minutes.
Analyses are done by comparing the effects of the training period (T_c) with those of the waiting period (W_c). First results stem from 33 subjects (26 females, 7 males) in the training condition and 32 subjects (23 females, 9 males) in the waiting condition. The training induced a significant improvement in subtraction (HRT and AC), while no improvement was found after the waiting period (Tab. 2). Pre-tests showed no significant difference between the groups.

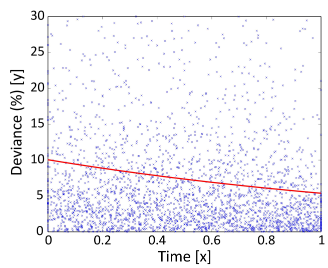
	Cond.	**Pre-Score(SD)**	**Post-Score(SD)**	**sig.**	**Comparison**
HRT	T_c	12.9 (5.38)	16.7 (5.3)	1.5e-8	2.6e-5 (2.9e-5)
	W_c	14.84 (6.47)	15.06 (5.87)	0.72	
AC	T_c	50.53 (27.25)	60.63 (26.3)	4.5e-4	1.9e-3 (2.0e-3)
	W_c	55.18 (25.24)	52.9 (27.74)	0.42	

Table 2. Comparison of test improvement between training and waiting condition. The last column shows the results of a t-test on the improvements assuming same variance and different variances, respectively.

The improvement is supported by additional evidence: the percentage of training time children spent with subtraction tasks. In fact, 62% (73% if considering key skills only) of arithmetical tasks consist of subtractions. The focus on subtraction and the significant improvement coming with it is scientifically interesting as performance in subtraction is considered the main indicator for numerical understanding [12]. Consistently with this, improved number line representation is directly measurable from the input data. Over time, children achieved greater accuracy when giving the position of a number on a number line (Fig. 8). The analysis of the accuracy is performed using a NLME model:

$$y_i \sim \text{Poisson}(\lambda_i) \text{ with } \lambda_i = e^{b_0 + b_1 \cdot x_i + u_i} \text{ and } u_i \sim \mathcal{N}(0, \sigma^2) \tag{4}$$

where u_i denotes the noise term, x_i the normalized sample indices ($x_i \in [0, 1]$) and y_i the deviance. Fitting is performed using one group per user.



	$\mathbf{b_0}$	$\mathbf{b_1}$
Estimate(SD)	2.3 (0.07)	-0.63 (0.02)
sig.	<1e-4	<1e-4
95% ci	[2.17 2.44]	[-0.67 -0.58]

Fig. 8. Landing accuracy in the range 0-100 increases over time (left). Exact coefficients of NLME along with standard deviation (in brackets) are plotted by respective significance (sig.) and confidence intervals (ci).

5 Conclusion

This study introduces a model of the cognitive processes of mathematical development based on current neuropsychological findings. Experimental results demonstrate that domain knowledge is well represented by dynamic Bayesian networks. The predictive model enables the optimization of the learning process through controlled cognitive stimulation. Regression analysis highlights sustained improvement; in particular, the possibility to go back significantly (and rapidly) reduces the error rate and yields an overall increased learning rate. Results are validated by large-scale input data analysis as well as external measures of effectiveness. The student model has the potential to be further refined by incorporating available experimental data.

Acknowledgments. We thank B. Solenthaler for helpful suggestions. The work was funded by the CTI-grant 11006.1 and the BMBF-grant 01GJ1011.

References

1. Baschera G.-M., Gross M.: Poisson-Based Inference for Perturbation Models in Adaptive Spelling Training. Int. J. of Artificial Intelligence in Education 20 (2010)

2. Shalev R. & von Aster M.G.: Identification, classification, and prevalence of developmental dyscalculia. Enc. of Language and Literacy Development, 1–9 (2008)
3. Wilson A.J., Dehaene S., Pinel P., Revkin S.K., Cohen L., Cohen D.: Principles underlying the design of "The Number Race", an adaptive computer game for remediation of dyscalculia. Behavioral and Brain functions 2:19 (2006)
4. Butterworth B., Varma S., Laurillard D.: Dyscalculia: From Brain to Education. Science 332, 1049 (2011)
5. Kucian K., Grond U., Rotzer S., Henzi B., Schönmann C., Plangger F., Gälli M., Martin E., von Aster M.: Mental Number Line Training in Children with Developmental Dyscalculia. NeuroImage 57(3), 782–95 (2011)
6. Koedinger K. R., Anderson J. R., Hadley W. H., Mark M. A.: Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education 8(1), 30-43 (1997)
7. Mislevy, R.J., Almond, R.G., Yan, D., & Steinberg, L.S.: Bayes nets in educational assessment: Where do the numbers come from? Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, 437-446 (1999)
8. Rau M.A., Aleven V., Rummel N.: Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. Proc. of the 14th Int. Conf. on Artificial Intelligence in Education, 441-448 (2009)
9. García C.E., Prett D.M., Morari M.: Model predictive control: theory and practice. Automatica 25 (3): 335–348 (1989)
10. Busetto A.G., Buhmann J.M.: Structure Identification by Optimized Interventions. Journal of Machine Learning Research Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics, 57–64 (2009)
11. Baschera G.-M., Busetto A.G., Klingler S., Buhmann J.M., Gross M.: Modeling Engagement Dynamics in Spelling Learning. LNCS Proc. of the 15th Int. Conf. on Artificial Intelligence in Education, 31–38 (2011)
12. Dehaene S.: The Number Sense: How the Mind Creates Mathematics. Oxford University Press (2011)
13. Dehaene S.: Varieties of numerical abilities. Cognition 44, 1–42 (1992)
14. von Aster M.G., Shalev R.: Number development and developmental dyscalculia. Developmental Medicine and Child Neurology 49, 868–873 (2007)
15. Ostad S.A.: Developmental differences in addition strategies: A comparison of mathematically disabled and mathematically normal children. British Journal of Education Psychology 67, 345–357 (1997)
16. Ostad S.A.: Developmental progression of subtraction strategies: A comparison of mathematically normal and mathematically disabled children. European Journal of Special Needs Education 14, 21–36 (1999)
17. Kucian K. & Kaufmann L.: A developmental model of number representation. Behavioral and Brain Sciences 32:313 (2009)
18. Friedmann N., Murphy K., Russell S.: Learning the Structure of Dynamic Probabilistic Networks. Uncertainty in AI (1998)
19. Mooij J.M.: libDAI: A free & open source C++ library for Discrete Approximate Inference in graphical models. J. of Machine Learning Research 11, 2169–2173 (2010)
20. Pinheiro J.C., Bates D. M.: Approximations to the Log-likelihood function in Nonlinear Mixed-Effects Models. Journal of Computational and Graphical Statistics 4:1, 12–35 (1995).
21. Breiman L.: Bagging predictors. Machine Learning 24 (2): 123–140 (1996)
22. Haffner J., Baro K., Parzer P., Resch F.: Heidelberger Rechentest (HRT): Erfassung mathematischer Basiskompetenzen im Grundschulalter. Hogrefe Verlag (2005)